

## Grundlagen Semantic Web

Seminar für Computerlinguistik, Universität Heidelberg

Sebastian Rudolph

Wintersemester 2008/09

<http://semantic-web-grundlagen.de>

### Übung 2: RDF und RDF Schema

---

**Aufgabe 2.1** (Übernommen aus Übung 1) XML Schema bietet eine Vielzahl vordefinierter Datentypen. Genaue Informationen über diese Typen können leicht online in den offiziellen W3C-Dokumenten nachgelesen werden <http://www.w3.org/TR/xmlschema-2/>. Verwenden Sie dieses Dokument zur Beantwortung der folgenden Fragen, und machen Sie sich dazu zunächst grob mit der Struktur der Spezifikation vertraut. Verwenden Sie Textsuche und Inhaltsverzeichnis um gezielt Informationen zu finden – keinesfalls sollten Sie die gesamte Spezifikation durchlesen!

- (a) Informieren Sie sich über die Datentypen `xsd:decimal`, `xsd:dateTime`, und `xsd:anyURI`, so dass Sie jeweils grob in eigenen Worten den lexikalischen Bereich und Wertebereich beschreiben können. Geben Sie jeweils ein (nicht schon in der Spezifikation verwendetes) Beispiel für einen syntaktisch korrekten Wert dieser Typen an.
- (b) XML Schema definiert eine Reihe von numerischen Datentypen. Informieren Sie sich über die Datentypen `xsd:int`, `xsd:decimal`, und `xsd:float`. Geben Sie für jeden dieser Datentypen einen möglichst charakteristischen Beispielwert an.
- (c) Worin besteht der Unterschied zwischen den Datentypen `xsd:int` und `xsd:integer`?

**Aufgabe 2.2** Erklären Sie die folgenden Elemente des Vokabulars von RDF und RDF Schema (ausgehend von den üblichen Namensraumdeklarationen).

- |                                |                               |
|--------------------------------|-------------------------------|
| (a) <code>rdf:type</code>      | (f) <code>rdf:resource</code> |
| (b) <code>rdf:about</code>     | (g) <code>rdf:nil</code>      |
| (c) <code>rdf:Property</code>  | (h) <code>rdfs:label</code>   |
| (d) <code>rdf:Seq</code>       | (i) <code>rdf:member</code>   |
| (e) <code>rdfs:Resource</code> | (j) <code>rdf:value</code>    |

**Aufgabe 2.3** Betrachten Sie die folgende RDF-Repräsentation:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:ex="http://example.org/">

  <rdf:Description rdf:about="http://example.org/deutschland">
    <rdf:type rdf:resource="http://example.org/Land"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/hauptstadt_von">
    <rdf:type rdf:resource=
      "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
    <rdfs:domain rdf:resource="http://example.org/Stadt"/>
    <rdfs:range rdf:resource="http://example.org/Land"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/Land">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:label xml:lang="en">country</rdfs:label>
  </rdf:Description>

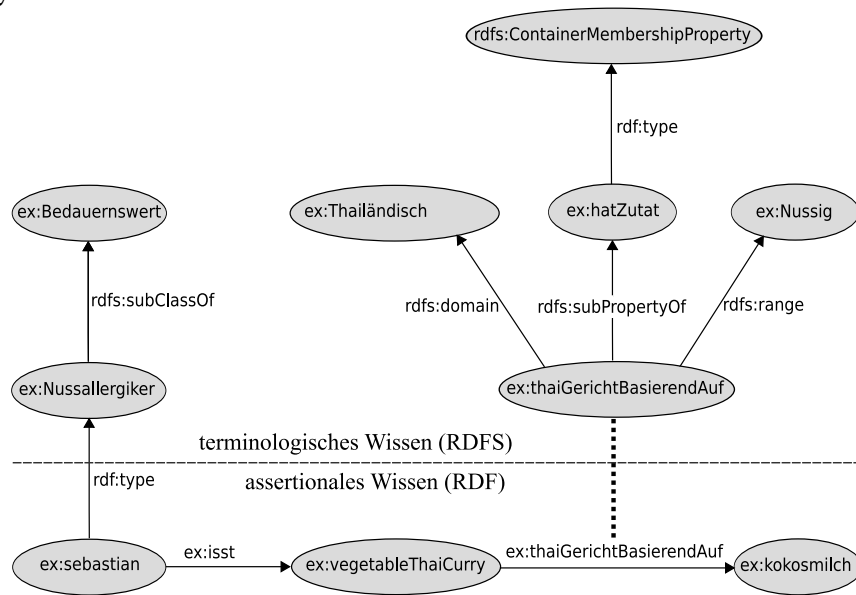
  <rdf:Description rdf:about="http://example.org/berlin">
    <rdfs:label xml:lang="en">Berlin</rdfs:label>
    <rdf:type rdf:resource="http://example.org/Stadt"/>
    <ex:hauptstadt_von rdf:resource="http://example.org/deutschland"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/Stadt">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:label xml:lang="en">city</rdfs:label>
  </rdf:Description>

</rdf:RDF>
```

- (a) Beschreiben Sie in eigenen Worten, was durch diese Repräsentation ausgedrückt wird.
- (b) Erstellen Sie eine graphische Darstellung der obigen RDF-Repräsentation.
- (c) Übersetzen Sie das gegebene RDF/XML-Dokument in die Turtle-Syntax.

**Aufgabe 2.4** Stellen Sie die folgende vereinfachte Darstellung eines RDF-Graphen mit Hilfe der RDF/XML-Syntax dar:



**Aufgabe 2.5** Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind:

- Leere Knoten können für beliebige Ressourcen stehen.
- URIs können für beliebige Ressourcen stehen.
- Jeder leere Knoten hat eine ID.
- Zwei leere Knoten mit unterschiedlicher ID können für die gleiche Ressource stehen.
- Zwei unterschiedliche URIs können für die gleiche Ressourcen stehen.
- Kommen in mehreren RDF-Dokumenten leere Knoten mit der selben ID vor, dann müssen sie für die selbe Ressource stehen.
- Kommen in mehreren RDF-Dokumenten die gleichen URIs vor, dann müssen sie für die selbe Ressource stehen.
- Zwei unterschiedliche Literale können niemals für den gleichen Wert stehen.
- Zwei Literale unterschiedlichen Datentyps können niemals für den gleichen Wert stehen.
- Eine URI kann niemals für den Wert eines Datentyps stehen.
- Leere Knoten können nicht als Prädikate in Tripeln auftreten.
- Leere Knoten können nicht für Propertys stehen (also für Ressourcen, welche der Klasse `rdf:Property` angehören).

**Aufgabe 2.6** Erklären Sie den Unterschied zwischen offenen und geschlossenen Listen, und geben Sie jeweils ein Beispiel in Turtle-Syntax an. Was ist mit „offen“ und „geschlossen“ gemeint (außer dem technischen Begriff)?

**Aufgabe 2.7** Stellen Sie folgende Aussagen mithilfe reifizierter Tripel grafisch dar (aus Platzgründen können Sie anstelle kompletter URIs eigene kurze Bezeichner für Elemente verwenden):

- (a) Romeo meinte, Julia sei tot.
- (b) John glaubt, dass Mary ihn heiraten möchte.
- (c) Der Zwerg bemerkte, dass irgend jemand von seinem Tellerchen gegessen hatte.

**Aufgabe 2.8** Entscheiden Sie, ob sich die folgenden Aussagen mittels RDF(S) zufriedenstellend modellieren lassen und geben Sie gegebenenfalls entsprechende RDF(S)-Spezifikationen an. Verwenden Sie dabei eine möglichst kompakte syntaktische Darstellung (auch bei der Wahl der URIs).

- (a) Jede Pizza ist eine Speise.
- (b) Pizzen haben immer mindestens zwei Beläge.
- (c) Jede Pizza der Klasse `PizzaMargarita` hat `Tomate` als Belag.
- (d) Alles, was einen Belag hat, ist eine Pizza.
- (e) Keine Pizza der Klasse `PizzaMargarita` hat Belag aus der Klasse `Fleisch`.
- (f) „Einen Belag haben“ ist eine Enthaltenseins-Beziehung.